Where, When & Which Concepts Does AlphaZero Learn? Lessons from the Game of Hex

Jessica Zosa Forde* and **Charles Lovering*** and **George Konidaris** and **Ellie Pavlick** and **Michael L. Littman**
{first}_{last}@brown.edu,
Brown University

Abstract

AlphaZero, an approach to reinforcement learning that couples neural networks and Monte Carlo tree search (MCTS), has produced state-of-the-art strategies for traditional board games like Chess, Go, and Hex. While researchers and game commentators have suggested that AlphaZero uses concepts humans consider important, it is unclear how these concepts are represented in the network. We investigate AlphaZero's representations in Hex using both model probing and behavioral tests. We find that the MCTS search initially finds important concepts, and then the neural network learns to encode these concepts. Concepts related to short-term end-game planning are best encoded in the final layers of the model, whereas concepts related to long-term planning are encoded in the middle layers of the model.

Introduction

Domain experts have observed that AlphaZero (Silver et al. 2016), a reinforcement learning approach that combines Monte Carlo tree search (Brügmann 1993) with neural networks, uses, but does not master, identifiable game concepts. For example, despite being exceptionally strong overall, AlphaZero appeared unable fully to project the implications of a *ladder*—a relatively simple concept in the game of Go (Tian et al. 2019).

An agent's good performance can mask flaws in neural network systems generally (Poliak et al. 2018b; Gururangan et al. 2018), and reinforcement learning systems in particular (Witty et al. 2018; Zhang, Wu, and Pineau 2018). By visualizing which features of the environment an agent relies upon, e.g., via saliency maps (Simonyan, Vedaldi, and Zisserman 2014) or attention (Mott et al. 2019), researchers can infer whether or not a reinforcement learning agent is using appropriate features, and therefore generalizes effectively. These techniques can help determine if the agent is behaving appropriately, say by attending to enemy sprites. However, it is difficult to interpret the agent's behavior as a whole from these findings because these methods work on a per image basis. Understanding how a model represents relevant concepts can better help us understand its decisions (Kim et al. 2018), and further, let us make predictions about its behavior (Lovering et al. 2021).

A characterization of which concepts an agent "understands" summarizes that agent's abilities, and where those abilities fall short. We use two techniques–*model probing* (Alain and Bengio 2017; Conneau et al. 2018; Poliak et al. 2018a; Marvin and Linzen 2018; Tenney, Das, and Pavlick 2019; Sinha et al. 2021) and *behavioral tests* (Linzen, Dupoux, and Goldberg 2016; McCoy, Pavlick, and Linzen 2020; Warstadt et al. 2020; Gauthier et al. 2020)–to interpret AlphaZero's behavior at a conceptual level. Probing classifiers measure if model activations can be used to discriminate between concepts: In natural language processing, they are used to determine if deep learning models encode linguistic concepts. These classifiers can be used analogously to determine if reinforcement learning agents encode tactical and strategic conceptual information (Anand et al. 2019; McGrath et al. 2021). However, probing performance alone is insufficient to determine that these concepts play an important role in decision making: Information may be encoded but not used (Lovering et al. 2021). We address this issue in our work by also examining agent behavior over game concepts. To do so we use behavioral tests, which evaluate an agent's decisions in a situation tailored to require the understanding of a specific concept. Using both probing classifiers and behavioral tests, we can study how internal representations and external behavior relate.

We leverage both of these concept-level evaluation methods to interpret AlphaZero (AZ) trained to play Hex. Hex is a board game similar to Go, which we introduce below. We probe for concepts that are traditionally taught to new players of Hex. Given a dataset of boards with and without a specific concept, like *bridge* (Figure 2a), we train a classifier over AZ's neural network activations to determine if the concept is encoded. We also test if the model behavior aligns with the expectations of this concept—a behavioral test. Applying these methods, we investigate how concepts are represented within AZ, when concepts are learned during training, and where concept are represented within AZ's neural network.

We analyze the top performing model from Jones (2021), and find that (1) concepts that pertain to short-term end-game planning are best represented in the final layers of the network, whereas concepts that pertain to long-term planning are best represented in earlier in the network; (2) concepts appear to originate with MCTS—with MCTS overrid-

*Equal contribution.

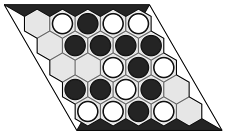


Figure 1: An example winning board from the game of Hex. To win, a player must use their pieces to form a connected chain between the edges matching their color. In this example, black connects the two black edges and wins. Hex boards also can be of larger height and width; we evaluate AlphaZero on a 9x9 board.

ing the neural network's policy prediction early in training—but later in training these concepts are incorporated directly into the model's network. Investigation into how concept representations causally impact resulting model behavior is a rich direction for future work.

Concepts in Hex

We study AlphaZero (AZ) agents trained to play Hex (Gardner 1961), a game where two players take turns filling cells until one player builds a chain across the board. Unlike Go, there are no captures; once a cell is filled with a piece, the pieces stays there for the remainder of the game. In Figure 1, white connects from left to right, and black from top to bottom. Hex cannot end in a tie (Gale 1979), and given perfect play, black, the first player, will win (Gardner 1961).[1]

In Hex, certain templates, i.e., patterns of cells, have been recognized as useful. While the properties of concepts are debated (Margolis, Laurence, and others 1999), here, in a board game setting, we consider a concept to be a useful template that generalizes across board configurations. For our purposes, "understanding" a concept amounts to recognizing it and leveraging its implications during gameplay.

Concept Taxonomy

From Seymour (2019) and King (2004), we identify nine concepts, summarized in Figure 2. Broadly, we categorize these concepts as being positive or negative. Positive concepts prescribe which actions to take, whereas negative concepts prescribe which actions not to avoid.

Positive Concepts With the goal of Hex being to build a chain across the board, it is helpful to recognize when cells are *virtually connected*, that is, even in response to perfect adversarial play, the cells are guaranteed to connect (Hayward et al. 2005; Pawlewicz et al. 2014). All the positive concepts favor the player that owns the concept on the board.

We use three types of positive concepts. Internal concepts are templates that appear within the interior of the board. The *bridge* (Figure 2(a)) is the simplest such concept. The larger internal templates – *crescent*, *trapezoid*, *span* (Figure 2(b,c,d)) – provide several possibilities to connect a player's pieces. *Edge* concepts virtually connect a single cell to an edge. Ladders in Hex are similar to ladders in Go. In Figure 2(g,h), black attempts to connect to the bottom edge. A *bottleneck* (Figure 2(g)) prevents black from connecting, whereas an *escape* (Figure 2(h)) allows black to connect.

Negative Concepts Negative concepts include empty cells that are to be avoided in play. *Dead* cells (Figure 2(i)) cannot impact the outcome of the game regardless of the player that fills the cell. It is often difficult to compute if a cell is dead (Bjornsson et al. 2006), but there are templates in which dead cells can be identified. If a player can make a cell dead, such as A in Figure 2(j)), it is *captured*.

Long-term vs Short-term Concepts

Concepts have different move implications depending on the condition of the board. We find that whether or not the concept is connected to the edge of the board has the most significant impact on AlphaZero's representations (Figure 5). If a concept is connected, it means the owner of that concept can win the game if it understands how to use that concept. For example, in Figure 4(d), the bridge is connected, and all that is required is a *short-term* understanding of the concept and the board. If a concept is disconnected, it still pertains to the *long-term* strategy.

AlphaZero's Gameplay

AlphaZero (AZ) uses both neural networks and MCTS to produce a policy. The neural network produces a value estimate of the board, and a policy distribution over which action to take next. AZ's MCTS creates an updated policy distribution based on roll-outs of possible next moves. Neither the network's loss, nor MCTS's structure, encourage winning quickly (Silver et al. 2017). AZ selects actions that most likely result in a win *regardless of the number of actions necessary to achieve that win*. This is in contrast to MoHex (Huang et al. 2014; Pawlewicz and Hayward 2014; Pawlewicz et al. 2015), a state of the art Hex agent, which is hard-coded to select the move that forms the shortest connection between the winning player's pieces at the end of the game (Arneson et al. 2018). Figure 3 shows an example of AZ delaying the game. These delays impact how we can test AZ's behavioral understanding of concepts. To determine if AZ uses a concept, we present AZ with a situation where using the concept is the only way to win. If AZ fails to do so, we can be confident that it does not understand the concept.

Probing Tasks and Behavioral Tests

To understand the concepts encoded within AZ, we probe its internal representations; to evaluate if these concepts are used, we test its behavior on tailored board configurations. Each task uses five replicate seeds. By evaluating AZ across checkpoints and network layers, we understand how and where the model recognizes these concepts. Specifically, we evaluate the top-performing agent trained by Jones (2021) across 21 training checkpoints. Our code and results are publicly available. Furthermore, we release example images of boards created for our probing classifiers and videos of the

[1] Hex is often played with a "swap rule" that makes the game more even between black and white. See Jones (2021), whose implementation we use, for further discussion on the swap rule which was not included to simplify the game implementation.

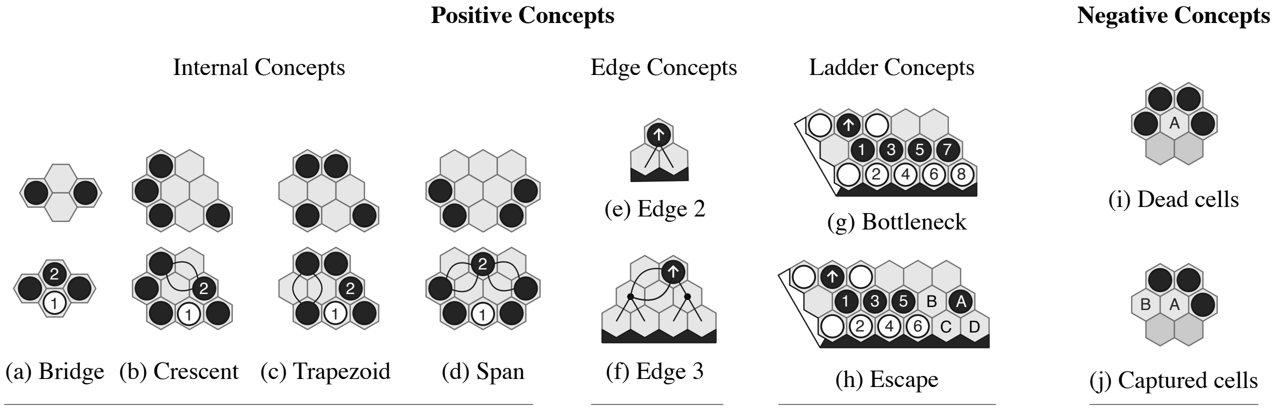


Figure 2: **Hex templates exemplifying game concepts.** Concepts within the game of Hex are patterns on the board formed by a player's pieces that have known strategic and tactical implications. Positive concepts provide the owner of the concept multiple ways to connect the pieces within the template together, despite possible attacks from the opponent. An example of these properties is the bridge (a). If white plays move 1, black still can connect between the two pieces of the bridge by playing move 2. Negative concepts are board structures that contain open cells that neither player should fill. For example, neither player should play move A in (i) because it cannot impact the outcome of the game. Arrows indicate that the piece is connected to the opposite side of the board; the lines show the bridge concept within the other concepts.

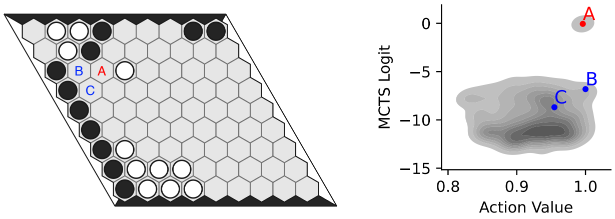


Figure 3: **Example end game board demonstrating AlphaZero's inefficiency of last moves.** Black is close to winning the game and must decide its next move. Move B ends the game. For each available move, we measure AZ's estimate of the action's value (Action Value) and its log probability of selecting each action using MCTS (MCTS Logit), averaged over 100 runs of MCTS. Both A and B have action values near 1: AlphaZero estimates both moves to result in a win for black. However, A (red) has the highest MCTS logit, and therefore highest probability of being selected. While move A allows AZ to win the game on the next round, it unnecessarily extends the game. Figure 7 shows that AZ plays unnecessary moves even at the end of training.

behavioral tests [2].

Representational Probing

Model probing measures how well a model's learned representations encode a concept (Alain and Bengio 2017). Probing entails training a linear classifier (the *probe*) over model activations to predict the presence of the desired concept. Thus, concepts are defined by a set of examples. The classifier's test performance is used to interpret how well the original model encoded the concept. We largely follow Tenney, Das, and Pavlick (2019): For a given board $H^{(0)}$, we extract activations $H^{(l)}, l \in 1..L$ for each neural network layer. We then train linear classifiers $\mathcal{P}^{(l)}, l \in 0..L$ per layer to predict the presence of a concept in the board, y; these classifiers are our concept probes.

It is important to compare probing results against a baseline. We follow Hewitt and Liang (2019)'s procedure to measure *concept selectivity*, the delta between probing accuracy over a concept and random control. To form the random control, for each board in the probing dataset $(H^{(0)}, y)$, we consistently map each cell in that board to a random cell, forming $H_s^{(0)}$. In this way, the same information is encoded in the original boards, but we expect the shuffled boards to be irrelevant to Hex. Next, we train a set of linear probes $\mathcal{P}_s^{(l)}, l \in 0..L$ over the control boards $H_s^{(0)}$ to predict y . Now, finally, we can compute the concept selectivity by finding the delta in test accuracy between $\mathcal{P}_s^{(l)}$ and $\mathcal{P}^{(l)}$.

The higher the selectivity, the greater the extent that the model encodes the concept above what could be explained by a baseline. In Figure 5, we report the highest probing accuracy and selectivity across layers.

Implementation Details. Each concept is defined by a set of boards with and without a concept. We train and evaluate probing classifiers over AZ's encoding of the boards. To generate boards for each concept, we translate the minimal templates across an empty board. Then, we add random enemy pieces to the board. Negative instances of a given concept match the statistics of the positive examples, except that the pieces related to the concept template are randomly moved across the board. This is the long-term version of a concept. For the short-term concept, we connect the template to the edges of the board. Each probing dataset

[2] The code, results and examples can be found here: `https://bit.ly/alphatology`

has 2500 positive and 2500 negative examples. We generate multiple probing datasets across a range of conditions: long-term vs short-term, black to play vs white to play, black with the concept vs white with the concept. Only long-term vs short-term conditions impact results.

Behavioral Tests

Where model probing asks if concepts are represented within the model, behavioral tests asks if the model knows how to use the concept in gameplay. We measure the how well AZ's uses each concept by the percent of tests for that concept which has passed. To interpret the behavioral tests, they must have clear behavioral expectations.

Because AlphaZero (AZ) does not win in a minimal number of moves, our behavioral tests for positive concepts (Figure 2) are forced: If AZ can understand the concept and play the expected moves, AZ wins and passes the test, otherwise AZ loses the game and fails the test. Success is necessary but not sufficient to establish that the model has the concept. A negative result means that AZ is entirely unable to use the concept, whereas a positive result means that AZ uses the concept to win games in forced situations.

For negative concepts (Figure 2), we have clear behavioral expectations. Dead and captured cells should never be filled. Thus, the behavioral test for negative cells checks that during a selfplay continuation of a board containing a dead (or captured) cell, the agent does not fill that cell.

Implementation Details. For each concept, we create behavioral tests that comprise a board, forcing moves, and expected moves. For the dead and captured cells there are no forcing or expected moves, only the moves to avoid. Figure 4 is a visual example of how we generate behavioral tests. To generate 100 behavioral tests, we translate the concept templates to a sampled valid board position. We then connect the concept to the edges of the board. In self-play, there are often multiple relevant regions on the board that contribute to a win or loss. However, by focusing the area of gameplay to a single focal point, we can deliberately test if AZ learns to use the given concept. To focus the area of gameplay, we add connections for the other player up to the region of the concept *such that the other player wins if the agent fails to pass the behavioral test*. Finally, we add the appropriate number of random pieces – which do not meaningfully impact the state of the game – to ensure that the board position is valid.

Limitations

Evaluating AZ in a simple (and cheap) setting that has been solved (Arneson, Hayward, and Henderson 2011) and for which perfect-play baselines exist (Huang et al. 2014), makes it easier to interpret and guide future work. Expanding to other games will prove which trends are particular to our setting, and which generalize elsewhere. While we find a consistent relationship between the probing and behavioral tests, we do not run a counterfactual study. For example, we do not show that mistakes in recognizing a concept on a given board, lead to mistakes to use that concept. Future research to design such studies would add to this line of work.

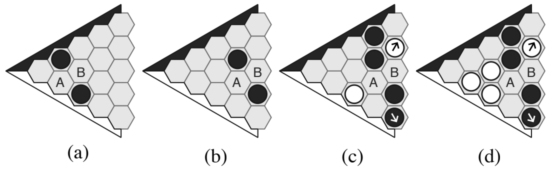


Figure 4: **Creating behavioral tests from concept templates.** We build synthetic boards that test if the agent can use a given concept to win the game. Each concept contains a minimal template representative of a board concept. In this example, we demonstrate the building of a behavioral test for the bridge (Figure 2a). The minimal template (a) is translated to a random position on the board (b). Then both players' pieces are connected to their respective edges they need to utilize to win the game (c). Finally, noise pieces are added to form a valid board (d). Cells A, B are used to define the behavioral test. If white plays A, black must play B to win the game. (Only half a 5x5 board is shown for space.)

Results

To understand which concepts AZ learns, we examine if its neural network encodes the concepts (probing tests) and if AZ can use the concepts to win games (behavioral tests).

AlphaZero Recognizes and Uses Concepts

Figure 5 shows that AZ successfully encodes short-term concepts with high selectivity scores. The long-term concept scores are also learned with slightly lower scores. By the end of training, AZ uses all positive concepts to win (Figure 6).

We measure the rate at which the policy network and MCTS recommend the correct action. Passing rates for the policy network and MCTS are above zero at halfway through training. MCTS passing rates increase slightly before policy network passing rates increase (Figure 6). These differences in passing rates between MCTS and the policy network suggest that conceptual information is incorporated

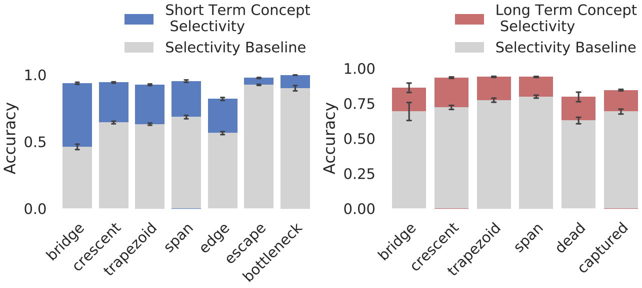


Figure 5: **AlphaZero successfully encodes short-term and long-term concepts.** The concept selectivity, indicated by the color bars, is the accuracy of a probe trained to identify a concept, minus the accuracy of a probe trained over a random control baseline. The random control randomly maps board cells so that the same information is encoded, but the random mapping should be irrelevant to Hex. We report selectivity based on the layer with the highest test accuracy.

from MCTS into the policy network. Because the policy network is trained to match the outputs of MCTS, similar passing rates are expected.

We additionally record how often the correct action's log probability is more than one standard deviation above the mean (z-score > 1). Interestingly, the relative magnitude of the the correct action increases at the start of training, long before the passing rates improve. Thus, it seems that there is some "pre-conceptual" information learned, which results in a large increase in passing rates 60% through training . Below, we investigate if this "pre-conceptual" information pertains to the structure of the board.

AZ improves behaviorally upon negative concepts but does not reach a 100% passing rate. Probing performance for the negative concepts, shown in Figure 5, is lower than for other concepts. These results align with evidence that AZ is prone to waste moves, and suggest that AZ's loss function may impact its learning of negative concepts.

Improvements in Behavioral Tests occur before Improvements in Probing Accuracy

We measure the the order of improvements in our behavioral and probing tasks, recording the checkpoint at which each task begins to improve and converges. We define a first improvement as the checkpoint at which the average accuracy/passing increases by a threshold from the baseline performance. In order to have the thresholds proportional to the range each metric, our threshold is 5% for behavioral tests and 2.5% for probing accuracy. Internal concepts meet our threshold for improvement at similar checkpoints for both behavioral and probing tasks. For each concept, our behavioral tests begin to improve before our probing tasks, which suggests that interaction with the concept is necessary to encoding it. Again, this finding is consistent with the structure of the loss function of AZ, which encourages the policy network output to match MCTS.

Concepts are Absorbed into the Model

Concepts are initially discovered (per our behavioral tests) via MCTS, before being predicted by the policy head of the agent. We find an analogous pattern in the concept representations. Figures 9 and 10 highlight which layer best represents the concept depends on whether it is a short or long-term concept. Long-term concepts, by the end of training, are best represented in the middle layers. Short-term concepts, throughout training, are best represented in the upper layers of the network. The layer that best represents the network is not the only layer that performs well in our probing task. Figure 9 shows the test accuracy of our probes for short-term bridges. Although the last layers perform best, even the second layer encodes the bridges well.

Cell Embeddings Capture Board Structure

Understanding Hex's concepts requires understanding the board's structure, i.e., which cells connect to which other cells. AlphaZero (AZ), with its feed forward network architecture, does not *a priori* represent this structure. Figure 6, above, shows evidence that some information is learned before the model is able to use the concepts. A possibility is that AZ spends the initial portions of training building up a representation of the board, and then uses this representation

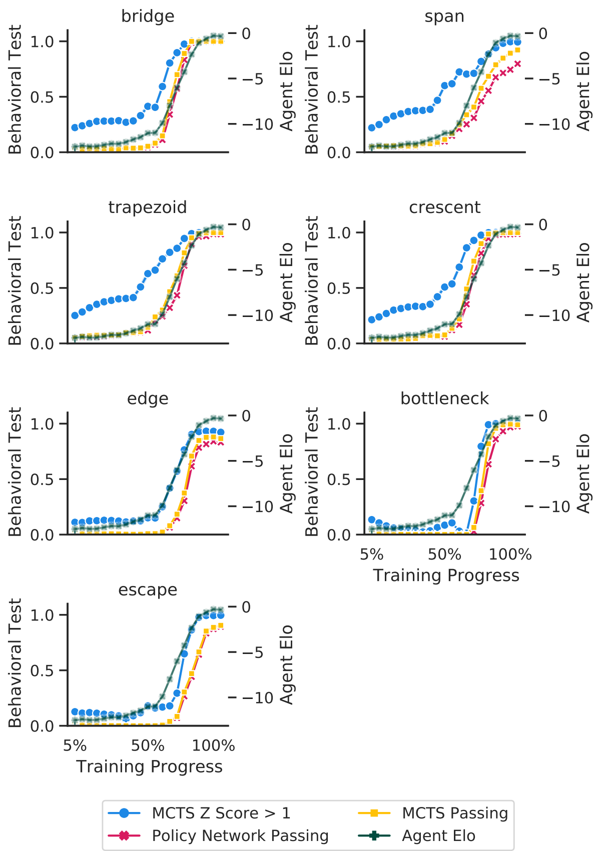


Figure 6: **AlphaZero learns to use the positive concepts.** At each checkpoint, we present AlphaZero with a set of example boards that test its ability to use each concept. MCTS (yellow) and the policy network (red) select actions that pass our behavioral tests with increasing frequency throughout training. We additionally report the rate at which the action that passes our behavioral test is one standard deviation above the mean (z score $>$ 1) (blue). The Agent Elo (dark green) measures AZ's general gameplaying ability; it increases as AlphaZero starts to use the concepts.

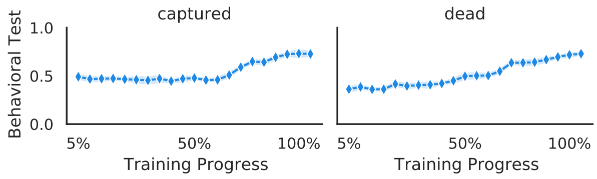


Figure 7: **AlphaZero does not fully "use" the negative concepts.** To pass these behavioral tests, AlphaZero must *avoid* playing cells on the board associated with the dead and captured concepts throughout a selfplay rollout. After being fully trained, it still plays these wasted moves in 25% of our behavioral tests.

to better play the game. However, neighborhood structure is learned later at the same time as the other concepts.

We investigate if the structure of Hex's board is represented in AZ's first layer using a structural probe (Hewitt and Manning 2019). A structural probe tests the relationship between neural network activations (or weights). We extract a cell embedding for each cell from the first layer of AZ.[3] From here, we compute the dot-products between each cell embedding. The dot-product score between ground-truth neighbors increases throughout AZ's training.

Qualitatively, the recovered structure is similar to the ground truth structure (Figure 11). The last row of Figure 8, *neighbors*, shows that improvements on this first concept occur at a similar time as to other concepts. The threshold for the initial improvement and convergence is 0.01 NCDG.

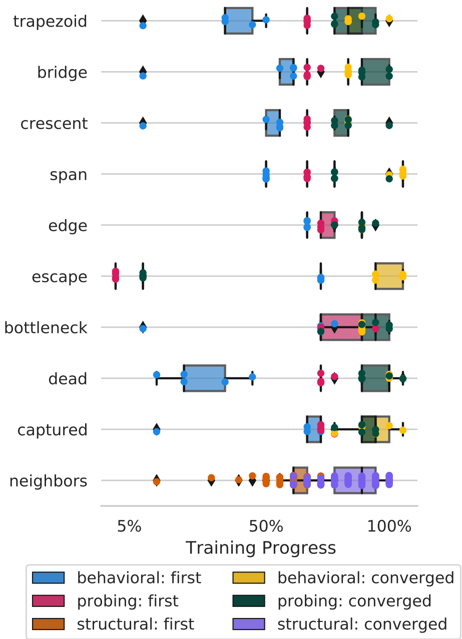
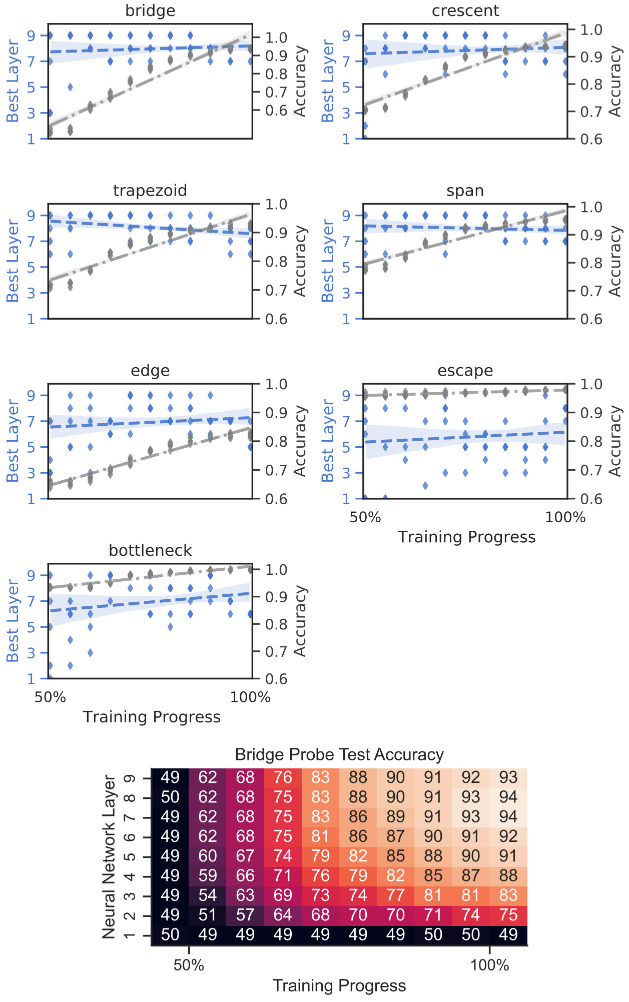


Figure 8: **Improvements in behavioral tests occur before improvements in probing accuracy.** This boxplot shows changes in concept representation and use during training. The checkpoint at which behavioral tasks improve are blue, and converge are yellow. The checkpoint at which probing tasks improve are pink, and converge are green. For internal concepts, the first four rows, the behavioral test improves before the probing accuracy. Probes to detect ladder escape concept (Fig. 2h) converge early in training, but the behavioral tests only improve late in training. Improvements in board structure are orange, and convergence, purple.



9	49	62	68	76	83	88	90	91	92	93
8	50	62	68	75	83	88	90	91	93	94
7	49	62	68	75	83	86	89	91	93	94
6	49	62	68	75	81	86	87	90	91	92
5	49	60	67	74	79	82	85	88	90	91
4	49	59	66	71	76	79	82	85	87	88
3	49	54	63	69	73	74	77	81	81	83
2	49	51	57	64	68	70	70	71	74	75
1	50	49	49	49	49	49	49	50	50	49

Figure 9: **Short-term concepts are best represented in the upper layer. Top**: As probe accuracy improves (gray), the layer with the highest accuracy (blue) stays in the upper layers of the model, as highlighted by the regression line. Error bars are one standard deviation above and below the mean estimate. **Bot:** This heatmap disaggregates the data in the bridge lineplot (top left). At the end of training, bridge is best represented in the top layers and well represented in the lower layers.

Related Work

Applying MCTS and neural networks to the Game of Hex. Jones (2021), whose trained agents we use, studied scaling laws between various parameters, finding a relationship between compute, board size, and desired agent performance. Where Gao, Hayward, and Müller (2017) and Takada, Iizuka, and Yamamoto (2017) both use value functions and MCTS to play Hex, Huang et al. (2014) (MoHex) combine connection detection, pattern matching, and MCTS, solving Hex for the 9x9 grid.

AlphaZero, anecdotally, uses concepts. Domain experts

[3] AlphaZero's first layer takes a flattened 1-hot encoding of the board and matrix multiplies this with a weight matrix. The embedding of each cell can be spliced out of that weight matrix. We only extract the cell embeddings for the current player.

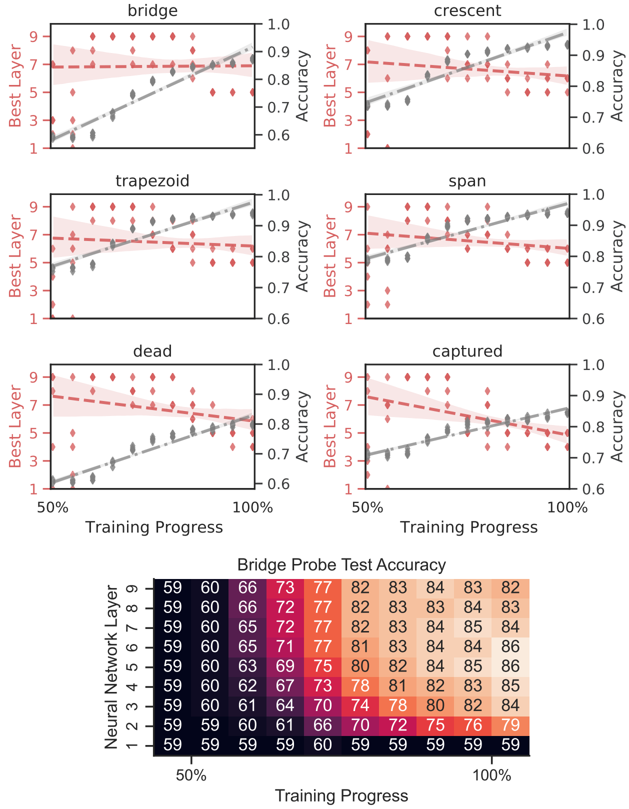


Figure 10: **Long-term concepts are best represented in middle layers**. As the accuracy of the probes trained to recognize each concept improves (gray), the layer with the highest accuracy (red) falls. Dashed lines show the linear regression of each metric, shading indicates one standard deviation above and below the mean estimate. **Bot:** This heatmap disaggregates the data in the bridge lineplot (top left).

have observed that AlphaGo (Silver et al. 2016) and related agents that combine Monte Carlo tree search (Brügmann 1993) with deep reinforcement learning use identifiable concepts within board games. In the commentary of AlphaGo's matches against Lee Sedol, Michael Redmond identified several common gameplay concepts demonstrated by AlphaGo (DeepMind 2016). Silver et al. (2018) noted that common *joseki* were used by AlphaZero during self-play. Tian et al. (2019) noted that Elf OpenGo only partially mastered ladder sequences within the game. Chess commentator Antonio Radić detailed how AlphaZero used *zugzwang* (Winter 1997) in the course of defeating Stockfish (Romstad et al. 2008; Radić 2017). Experts have already incorporated some of AlphaZero's innovations into their play (Nielsen 2019; Sadler and Regan 2019).

Probing neural networks for linguistic concepts. A wide range of linguistic concepts have been detected in NLP models using probes (Conneau et al. 2018; Poliak et al. 2018a; Marvin and Linzen 2018; Tenney et al. 2019; Hewitt and Manning 2019). There has been some discussion on what exactly good probing accuracy signifies. Hewitt and Liang (2019) calls for baseline controls, and to measure the gain in accuracy compared to these baselines.

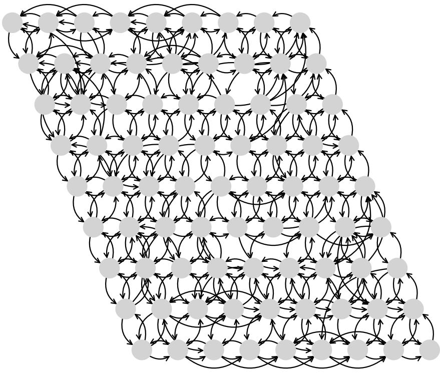


Figure 11: **Dot-product scores between cell embeddings recover board structure.**

Understanding reinforcement learning agents trained to play board games. Anand et al. (2019) compared unsupervised encoders' ability to represent various features of the state (e.g., number of opponent sprites). Sadler and Regan (2019) describes the impact of AlphaZero's gameplay on professional chess. Concurrent to our work, McGrath et al. (2021) also looks at how AlphaZero acquires game knowledge (which we term concepts) in chess. Using similar probing techniques, and the original (and larger) AlphaZero, they find that AlphaZero learns to encode many of the prototypical chess concepts. Related, but not focused on model understanding persay, Jhamtani et al. (2018) collect an annotated set of chess games. This type of resource is similar to what is used by McGrath et al. (2021) for their probing task.

Discussion

Our analyses suggest that AlphaZero (AZ) both represents and uses concepts that humans consider important when playing Hex. Which layer in the network best represents a concept depends on context: short-term concepts that inform actions at the end of the game are best encoded in the upper layers of the model, whereas long-term concepts are best encoded deeper in the network. AZ does not win in a minimal number of moves, often wasting moves once it reaches a secure position. This phenomena may explain why negative concepts are not as well encoded and used.

Combining both representational and behavioral approaches to analyze reinforcement learning agents allows for a fuller understanding of how they learn. Studying the representations of concepts allows us to ask (and answer) a rich set of questions about where that concept resides, and how it compares to other concepts. Studying the behavior of an agent on a given concept tests that this agent uses this concept; probing alone may be misleading. Behavioral tests can also expose heuristics the model may be using. In future work, finding the causal mechanisms of how an agent represents a concept and how it uses that concept will further illustrate how AlphaZero understands gameplay.

References

[2017] Alain, G., and Bengio, Y. 2017. Understanding intermediate layers using linear classifier probes. *ICLR Workshops*.

[2019] Anand, A.; Racah, E.; Ozair, S.; Bengio, Y.; Côté, M.-A.; and Hjelm, R. D. 2019. Unsupervised state representation learning in atari. *arXiv preprint arXiv:1906.08226*.

[2018] Arneson, B.; Henderson, P.; Pawlewicz, J.; Huang, A.; Young, K.; and Gao, C. 2018. Benzene. `https://github.com/cgao3/benzene-vanilla-cmake/blob/d450c01eb38803b1766ed9abea51568c4672f46b/src/hex/EndgameUtil.cpp`.

[2011] Arneson, B.; Hayward, R. B.; and Henderson, P. 2011. Solving hex: Beyond humans. In *Computers and Games*, 1–10. Springer Berlin Heidelberg.

[2006] Bjornsson, Y.; Hayward, R.; Johanson, M.; and van Rijswijck, J. 2006. Dead cell analysis in hex and the shannon game. 45–59.

[1993] Brügmann, B. 1993. Monte Carlo go. Technical report, Max Planck Institute of Physics.

[2018] Conneau, A.; Kruszewski, G.; Lample, G.; Barrault, L.; and Baroni, M. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2126–2136. Melbourne, Australia: Association for Computational Linguistics.

[2016] DeepMind. 2016. Match 1—Google DeepMind challenge match: Lee Sedol vs AlphaGo.

[1979] Gale, D. 1979. The game of hex and the brouwer Fixed-Point theorem. *Am. Math. Mon.* 86(10):818–827.

[2017] Gao, C.; Hayward, R.; and Müller, M. 2017. Move prediction using deep convolutional neural networks in hex. *IEEE Transactions on Games* 10(4):336–343.

[1961] Gardner, M. 1961. *The 2nd Scientific American Book of Mathematical Puzzles and Diversions*. Simon & Schuster.

[2020] Gauthier, J.; Hu, J.; Wilcox, E.; Qian, P.; and P. Levy, R. 2020. Syntaxgym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 70–76.

[2018] Gururangan, S.; Swayamdipta, S.; Levy, O.; Schwartz, R.; Bowman, S. R.; and Smith, N. A. 2018. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*.

[2005] Hayward, R.; Björnsson, Y.; Johanson, M.; Kan, M.; Po, N.; and Van Rijswijck, J. 2005. Solving 7× 7 hex with domination, fill-in, and virtual connections. *Theoretical Computer Science* 349(2):123–139.

[2019] Hewitt, J., and Liang, P. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2733–2743. Hong Kong, China: Association for Computational Linguistics.

[2019] Hewitt, J., and Manning, C. D. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4129–4138. Minneapolis, Minnesota: Association for Computational Linguistics.

[2014] Huang, S.-C.; Arneson, B.; Hayward, R. B.; Müller, M.; and Pawlewicz, J. 2014. MoHex 2.0: A Pattern-Based MCTS hex player. In *Computers and Games*, 60–71. Springer International Publishing.

[2018] Jhamtani, H.; Gangal, V.; Hovy, E.; Neubig, G.; and Berg-Kirkpatrick, T. 2018. Learning to generate move-by-move commentary for chess games from large-scale social forum data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1661–1671.

[2021] Jones, A. L. 2021. Scaling scaling laws with board games. *arXiv preprint arXiv:2104.03113*.

[2018] Kim, B.; Wattenberg, M.; Gilmer, J.; Cai, C.; Wexler, J.; Viegas, F.; et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, 2668–2677. PMLR.

[2004] King, D. 2004. Hall of hexagons. `https://www.drking.org.uk/hexagons/index.html`. Accessed: 2021-11-12.

[2016] Linzen, T.; Dupoux, E.; and Goldberg, Y. 2016. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics* 4:521–535.

[2021] Lovering, C.; Jha, R.; Linzen, T.; and Pavlick, E. 2021. Predicting inductive biases of pre-trained models.

[1999] Margolis, E.; Laurence, S.; et al. 1999. *Concepts: core readings*. Mit Press.

[2018] Marvin, R., and Linzen, T. 2018. Targeted syntactic evaluation of language models. *arXiv preprint arXiv:1808.09031*.

[2020] McCoy, R. T.; Pavlick, E.; and Linzen, T. 2020. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *57th Annual Meeting of the Association for Computational Linguistics, ACL 2019*, 3428–3448. Association for Computational Linguistics (ACL).

[2021] McGrath, T.; Kapishnikov, A.; Tomašev, N.; Pearce, A.; Hassabis, D.; Kim, B.; Paquet, U.; and Kramnik, V. 2021. Acquisition of chess knowledge in alphazero. *arXiv preprint arXiv:2111.09259*.

[2019] Mott, A.; Zoran, D.; Chrzanowski, M.; Wierstra, D.; and Rezende, D. J. 2019. Towards interpretable reinforcement learning using attention augmented agents. *arXiv preprint arXiv:1906.02500*.

[2019] Nielsen, P. H. 2019. The exciting impact of a game changer: When Magnus met AlphaZero. *New In Chess*.

[2014] Pawlewicz, J., and Hayward, R. B. 2014. Scalable parallel DFPN search. In *Computers and Games*, 138–150. Springer International Publishing.

[2014] Pawlewicz, J.; Hayward, R.; Henderson, P.; and Arneson, B. 2014. Stronger virtual connections in hex. *IEEE Transactions on Computational Intelligence and AI in Games* 7(2):156–166.

[2015] Pawlewicz, J.; Hayward, R.; Henderson, P.; and Arneson, B. 2015. Stronger virtual connections in hex. *IEEE Trans. Comput. Intell. AI Games* 7(2):156–166.

[2018a] Poliak, A.; Haldar, A.; Rudinger, R.; Hu, J. E.; Pavlick, E.; White, A. S.; and Van Durme, B. 2018a. Collecting diverse natural language inference problems for sentence representation evaluation. *arXiv preprint arXiv:1804.08207*.

[2018b] Poliak, A.; Naradowsky, J.; Haldar, A.; Rudinger, R.; and Van Durme, B. 2018b. Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*.

[2017] Radić, A. 2017. AlphaZero's "immortal zugzwang game" against Stockfish.

[2008] Romstad, T.; Costalba, M.; Kiiski, J.; and others. 2008. Stockfish.

[2019] Sadler, M., and Regan, N. 2019. Game changer. *AlphaZero's Groundbreaking Chess Strategies and the Promise of AI. Alkmaar. The Netherlands. New in Chess*.

[2019] Seymour, M. 2019. *Hex: A Strategy Guide*.

[2016] Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; Dieleman, S.; Grewe, D.; Nham, J.; Kalchbrenner, N.; Sutskever, I.; Lillicrap, T.; Leach, M.; Kavukcuoglu, K.; Graepel, T.; and Hassabis, D. 2016. Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484–489.

[2017] Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. *nature* 550(7676):354–359.

[2018] Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; Lillicrap, T.; Simonyan, K.; and Hassabis, D. 2018. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science* 362(6419):1140–1144.

[2014] Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop Track*. openreview.net.

[2021] Sinha, K.; Jia, R.; Hupkes, D.; Pineau, J.; Williams, A.; and Kiela, D. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. *arXiv preprint arXiv:2104.06644*.

[2017] Takada, K.; Iizuka, H.; and Yamamoto, M. 2017. Reinforcement learning for creating evaluation function using convolutional neural network in hex. In *2017 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, 196–201. IEEE.

[2019] Tenney, I.; Xia, P.; Chen, B.; Wang, A.; Poliak, A.; McCoy, R. T.; Kim, N.; Durme, B. V.; Bowman, S.; Das, D.; and Pavlick, E. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

[2019] Tenney, I.; Das, D.; and Pavlick, E. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4593–4601. Florence, Italy: Association for Computational Linguistics.

[2019] Tian, Y.; Ma, J.; Gong, Q.; Sengupta, S.; Chen, Z.; Pinkerton, J.; and Zitnick, L. 2019. Elf opengo: An analysis and open reimplementation of alphazero. In *International Conference on Machine Learning*, 6244–6253.

[2020] Warstadt, A.; Parrish, A.; Liu, H.; Mohananey, A.; Peng, W.; Wang, S.-F.; and Bowman, S. R. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics* 8:377–392.

[1997] Winter, E. 1997. Zugzwang. `https://www.chesshistory.com/winter/extra/zugzwang.html`. Accessed: 2021-11-12.

[2018] Witty, S.; Lee, J. K.; Tosch, E.; Atrey, A.; Littman, M.; and Jensen, D. 2018. Measuring and characterizing generalization in deep reinforcement learning. *arXiv preprint arXiv:1812.02868*.

[2018] Zhang, A.; Wu, Y.; and Pineau, J. 2018. Natural environment benchmarks for reinforcement learning. *arXiv preprint arXiv:1811.06032*.